

الف)

روش Join Merge-Sort یا Merge Join بر اساس مرتب‌سازی و ادغام عمل می‌کند. مراحل این روش عبارتند از:

1. مرتب‌سازی: دو جدول مورد نظر بر اساس ستون‌های پیوند مرتب می‌شوند.

2. ادغام: بعد از مرتب‌سازی، دو جدول به صورت خطی مرور می‌شوند و ردیف‌هایی که شرایط پیوند را برآورده می‌کنند با هم جفت می‌شوند.

روش Join Hash یا Hash Join بر اساس استفاده از جدول‌های هش عمل می‌کند. مراحل این روش عبارتند از:

1. ساخت جدول هش: یک جدول هش از یکی از جداول (معمولاً کوچکتر) بر اساس کلید پیوند ساخته می‌شود.

2. جستجو: جدول دیگر پیمایش می‌شود و هر رکورد آن در جدول هش جستجو می‌شود تا تطابق‌ها پیدا شوند.

مقایسه :

Join Merge-Sort:

○ زمانی که جداول از قبل بر اساس کلید پیوند مرتب شده‌اند.

- زمانی که استفاده از حافظه باید محدود باشد.
- برای پیوندهای تکراری و جداول با ایندکس‌های مناسب.

● Join Hash:

- زمانی که جداول بزرگ هستند و ایندکس ندارند.
- زمانی که حافظه کافی برای ساخت و نگهداری جدول‌های هش موجود است.
- برای پیوندهای تک‌مقدار و زمانی که برخوردهای هش به حداقل رسیده باشند.

(ب)

۱-

Single_Index Access Path:

اگر چند index داشته باشیم بهترین index انتخاب می شود

۲-

Multiple_Index Access Path:

اگر دو یا چند index قابل استفاده داشته باشیم، از اشتراک آنها استفاده می کنیم.

۳-

Sorted Index Access Path:

اگر لیست صفات groupby یک prefix ای از شاخص درختی باشد می توان نتیجه ای sort شده را برگرداند (Index Cluster).

۴-

Index Only Access Path:

اگر همه ی صفات در کلید شاخص باشند فقط کافی است از index استفاده کنیم.

(۱ج)

1. شاخص مناسب برای ستون های مورد نیاز وجود نداشته باشد.
2. تعداد رکوردها در جدول کم باشد و پویش کامل کارآمدتر باشد.
3. پرس و جو درصد زیادی از ردیف ها را برگرداند.
4. پرس و جو شامل توابع تجمیعی بدون شرط خاص باشد.
5. شرایط پیچیده یا توابع در شرط ها استفاده شده باشد.

6. زمانی که تعداد داده های fetch شده بخش قابل توجهی از کل داده ها باشد، اگر index نداشته باشیم یا index موجود بدرد ما نخورد از روش full table scan استفاده می شود.

۲الف)

الف)

تعداد صفحات R:

$$\left[100,000 \times 100 \times \frac{1}{4096} \right] = 2442$$

سپس کل R به 2442 صفحه جای می خورد. برای scan کردن باید 2442 صفحه را بخواند.

۲ب)

ب. R Sort فای : $2 \times 2 \times 2222 = 9741$

طرح درجہ

pass : $1 + \lceil \log_9 \lceil \frac{2222}{2} \rceil \rceil = 2$

S Sort فای : $2 \times 2 \times \lceil \frac{20000 \times 10}{2099} \rceil = 977 \times 2 = 1954$

pass

pass : $1 + \log \lceil \log_{19} \lceil \frac{20000 \times 10}{2099 \times 2} \rceil \rceil = 2$

Merge algorithm : $M+N = 2222 + 977 = 3209 \rightarrow total = 17095$

CS Scanned with CamScanner

ج ۲

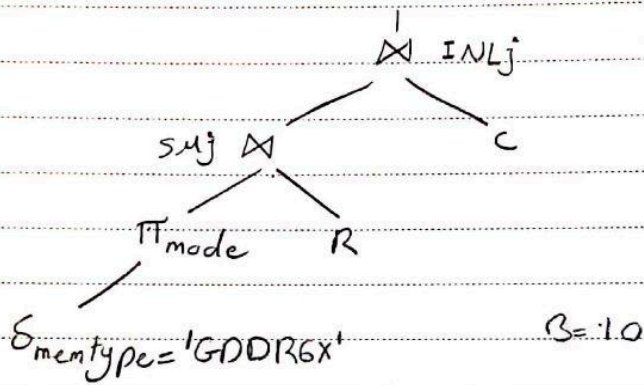
ج. R صفحات : $\frac{100,000 \times 100}{2099} = 2222 \text{ pages}$

S صفحات : $\frac{20,000 \times 10}{2099} = 977 \text{ pages}$

پس شرط تکرار نیست : $B \geq \sqrt{\frac{Mm(S,R)}{977}} \rightarrow 20 \neq 50$

PAPCO : $3(M+N) = 3(2222 + 977) = 10257$

CS Scanned with CamScanner



$$Customer = \frac{2800}{300} = 9.33 \quad Reat = \frac{110000}{400} = 275 \quad GPU = \frac{12000}{300} = 40$$

از پهنای سمت صاف: $\delta_{mentype}$ و $\delta_{mentype}$ به تبع ضرایب راست پس T_{mod} اعمال شده ابع ضرایب راست

در ادامه با SMj و R برای 450 به $3 = \lceil \log \left[\frac{450}{20} \right] \rceil + 1$ $paths$ هم مرتبه شود

$$1 + 2 \times 450 + 40 + 275 + 1 = 3192 \quad \leftarrow \text{رایانگی نهایی را به دست می آید}$$

توزیع $uniform$ بود و هر $1.5 = \frac{110000}{12000}$ از R ، $join$ شده پس

$$72000 = 18000 \times (3+1) \quad \text{هرینه}$$

که هر $18k$ یک تایل از C

هر 18000 تایل

Groupby یا یابی نیز برید که کوادر ما مرتب شده که بخاطر SMj بیشتر شده اند و کاری نمی تواند بینیم

$$75192 = 3192 + 72000 = \boxed{75192} \quad \text{هرینه کلی}$$

محاسبه هزینه‌های پرس و جو

فرضیات:

- رابطه Item دارای 5000 صفحه است.
- رابطه Store دارای 200 صفحه است.
- رابطه Transaction دارای 20,000 صفحه است.

1. انتخاب 'NYC' = S.city روی Store:

ابتدا انتخاب 'NYC' = S.city را انجام می‌دهیم چون یک شاخص روی S.city داریم.

هزینه انتخاب $\text{S.city} = 200 * 0.01 + \sqrt{(200 * 1/3 * 0.01)} + 2 = 5$

این انتخاب باعث می‌شود فقط 2 صفحه (10 تاپل) از Store باقی بماند.

2. Join رابطه‌های Transaction و Store با استفاده از INLJ با شاخص
:Clustered B+ Tree

با داشتن شاخص Clustered B+ Tree روی T.date و ارتفاع 2:

هزینه INLJ با شاخص Clustered B+ tree = تعداد تاپل‌های خارجی *
ارتفاع درخت

Transaction برای:

$$5000 = 50 * 10 = \text{تعداد تاپل‌های خارجی}$$

با انجام انتخاب روی '2004-12-15' = [T.date](#) به صورت on-the-fly:

$$100 = 5000 / 50 \text{ تاپل باقی می‌ماند.}$$

هزینه INLJ:

$$5000 = 50 * 10 * 10$$

و هزینه 20:

$$2 * 10 = 20$$

3. Join رابطه Item با Transaction با استفاده از INLJ با شاخص Hash

Unclustered:

با داشتن شاخص Hash Unclustered روی Item:

هزینه INLJ با شاخص Hash Unclustered = تعداد تاپل‌های

خارجی $\times (1 + \text{look-up})$

هزینه (1.2 lookup):

$$220 = (1 + 1.2) * 100$$

مجموع هزینه‌ها:

$$\text{هزینه کل} = 245 = 220 + 20 + 5$$

بنابراین، هزینه کل 245 عملیات I/O است.